Theoretical Computer Science 460 (2012) 16–25

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Reversible iterative graph processes[☆]

Mitre C. Dourado[a], Lucia Draque Penso[b], Dieter Rautenbach[b,*], Jayme L. Szwarcfiter[a]

[a] Instituto de Matemática, NCE, and COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 21941-590, Brazil
[b] Institut für Optimierung und Operations Research, Universität Ulm, Ulm, 89081, Germany

ARTICLE INFO

Article history:
Received 7 November 2011
Received in revised form 21 April 2012
Accepted 23 May 2012
Communicated by D. Peleg

Keywords:
Conquest and expansion games
Local majority process
Iterative polling process
Consensus
Dynamic monopoly
Reachability problems
Finite discrete dynamical systems
Local interaction games

ABSTRACT

Given a graph G, a function $f : V(G) \to \mathbb{Z}$, and an initial 0/1-vertex-labelling $c_1 : V(G) \to \{0, 1\}$, we study an iterative 0/1-vertex-labelling process on G where in each round every vertex v changes its label if and only if at least $f(v)$ neighbours have a different label. For special choices of the values of f, such processes model consensus issues and have been studied under names such as local majority processes or iterative polling processes in a large variety of contexts especially in distributed computing. Our contributions concern computational aspects related to the minimum cardinality $\mathrm{r}_f(G)$ of sets of vertices with initial label 1 such that during the process on G all vertices eventually change their label to 1. Such sets are known as dynamic monopolies or dynamos for short. We establish a hardness result and describe efficient algorithms for restricted instances on paths and cycles.

1. Introduction

We study iterative 0/1-vertex-labelling processes on finite, simple, and undirected graphs. Such processes have been studied in a variety of distinct areas such as social influence [8,13,18,32–34], neural networks [15], gene expression networks [17], immune systems [1], cellular automata [2], percolation [3], marketing strategies [9,10,18], finite discrete dynamical systems [4,29,35], local interaction games [23,25], and especially in distributed computing [11,12,16,20,26,30].

Formally, given a graph G, a *process on* G is an infinite sequence $\mathcal{P} = (c_t)_{t\in\mathbb{N}} = (c_1, c_2, \ldots)$ of labellings $c_t : V(G) \to \{0, 1\}$ of its vertices with the two labels 0 and 1. We consider processes $\mathcal{P} = (c_t)_{t\in\mathbb{N}}$ such that for every $t \in \mathbb{N}$, the labelling c_{t+1} is obtained from the labelling c_t by applying a certain rule, which depends on individual threshold values of the vertices. Given a graph G, a threshold function $f : V(G) \to \mathbb{Z}$, and a labelling $c : V(G) \to \{0, 1\}$, let $\mathrm{R}_f(c)$ be the unique labelling $c' : V(G) \to \{0, 1\}$ of G that satisfies

$$\forall\, v \in V(G) : \big(|\{u \in N_G(v) \mid c(u) \neq c(v)\}| \geq f(v) \Leftrightarrow c'(v) = 1 - c(v)\big)$$

where $N_G(u)$ denotes the neighbourhood of u in G. In $\mathrm{R}_f(c)$ a vertex v changes its label if and only if at least $f(v)$ neighbours of v have a different label in c. If $\mathcal{P} = (c_t)_{t\in\mathbb{N}}$ is a process on G and $c_{t+1} = \mathrm{R}_f(c_t)$ for every $t \in \mathbb{N}$, then we call $\mathcal{P}$ an R_f*-process*

[☆] The results of this paper were presented as a brief announcement at DISC 2010: M.C. Dourado, L.D. Penso, D.Rautenbach, and J.L. Szwarcfiter, Brief Announcement: On Reversible and Irreversible Conversions, *Distributed Computing*, 24th *International Symposium* (*DISC* 2010), Cambridge, MA, USA, September 2010, LNCS 6343 (2010), 395–397.

* Corresponding author. Tel.: +49 7315023630; fax: +49 731501223630.
E-mail addresses: mitre@nce.ufrj.br (M.C. Dourado), lucia.penso@uni-ulm.de (L.D. Penso), dieter.rautenbach@uni-ulm.de (D. Rautenbach), jayme@nce.ufrj.br (J.L. Szwarcfiter).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2012.05.042

on G. Clearly, such processes are uniquely determined by G, f, and c_1. Whenever $f : V(G) \to \mathbb{Z}$ is such that $f(v) = k$ for every $v \in V(G)$, we replace the subscript 'f' simply by 'k'. For a function $g : D \to \mathbb{N}$ and a set $X \subseteq D$, let $g(X) = \sum_{x \in X} g(x)$.

A process $\mathcal{P} = (c_t)_{t\in\mathbb{N}}$ on some graph G *converges to* 1 if there is some $t_0 \in \mathbb{N}$ such that $c_t(v) = 1$ for every $v \in V(G)$ and every $t \in \mathbb{N}$ with $t \geq t_0$.

In the present paper we are mainly interested in the minimum number of vertices of label 1 that result in a process that converges to 1. Therefore, let $\mathrm{r}_f(G)$ denote the minimum value of $c_1(V(G))$ for some R_f-process $\mathcal{P} = (c_t)_{t\in\mathbb{N}}$ on G that converges to 1. We call some R_f-process $\mathcal{P} = (c_t)_{t\in\mathbb{N}}$ on some graph G also the *R_f-process of $c_1^{-1}(1)$ on G* where $c_1^{-1}(1)$ denotes the set $\{v \in V(G) \mid c_1(v) = 1\}$. Furthermore, if $\mathcal{P} = (c_t)_{t\in\mathbb{N}}$ converges to 1, then $c_1^{-1}(1)$ is called an *R_f-conversion set of G*.

An R_f-process on a graph G models iterative voting or updating mechanisms related to consensus problems. The *local majority processes* considered by Mustafa and Pekeč [26] coincide exactly with R_f-processes where $f(v) = \left\lfloor \frac{d_G(v)}{2} \right\rfloor + 1$ for every vertex $v \in V(G)$. Certain R_f-processes and many of their natural variants were proposed under names such as *local majority processes* or *iterative polling processes* in distributed computing [16,20,27,30,31] in order to model problems related to agreement and consensus, system-level diagnosis, distributed database management, quorum systems, self-stabilisation, and local mending. Periodicity properties and convergence rates of majority processes were investigated [15,24,32–34]. Conversion sets and the parameter $\mathrm{r}_f(G)$ are closely related to *dynamic monopolies* or *dynamos* for short [5,6,10–12,19,21,22,28,30,31]. The parameter $\mathrm{r}_f(G)$ can also be considered as a variant of reachability problems considered for finite discrete dynamical systems [4], because it equals the minimum integer r such that the all-1-vertex-labelling is reachable from an initial 0/1-vertex-labelling with exactly r vertices having label 1.

Our contributions concern computational aspects of this parameter.

We prove an NP-hardness result for $\mathrm{r}_2(G)$. While several deep complexity results on reachability problems in discrete dynamical systems are known (see [4] and the detailed discussion in that paper), the optimisation goal in the definition of $\mathrm{r}_2(G)$ leads to a variant of a reachability problem that is not covered by existing results. For hardness results concerning the irreversible version of these processes refer to [7,8].

Furthermore, we present an algorithm computing $\mathrm{r}_f(G)$ for certain paths and cycles. Several of the cited references concerned with bounds, exact values, and algorithms for dynamic monopolies in graphs also consider rather special topologies that were proposed for real life networks. The complexity of our algorithm illustrates how difficult the detailed analysis of the considered processes is.

2. R_f-processes

R_f-processes on graphs, which start from small conversion sets and converge to 1 typically display an intuitively plausible behaviour: at least one region of the graph contains sufficiently many elements of the conversion set to guarantee convergence in this region. Other regions contain less elements of the conversion set, which alone would not be sufficient to guarantee convergence. Within the graph the convergence propagates from region to region at a limited speed. If some region contains too few elements of the conversion set, then these might disappear until the convergence has arrived at this region. Therefore, each region must contain at least as many elements of the conversion set as to guarantee the existence of vertices with label 1 for some time. If G is a graph, $f : V(G) \to \mathbb{Z}$ is a function, and A and B are disjoint sets of vertices such that $|N_G(v) \cap B| \geq f(v)$ for every $v \in A$ and $|N_G(v) \cap A| \geq f(v)$ for every $v \in B$, then in the R_f-process of some set S on G with $A \subseteq S$ and $S \cap B = \emptyset$, the vertices in $A \cup B$ will constantly switch labels. While this construction guarantees the existence of vertices with label 1, it actually impedes conversion. The proof of the next result relies on a suitable modification of this construction.

Proposition 1. *If T is a tree of order n with l leaves, then $\mathrm{r}_2(T) \leq \frac{n+l}{2}$.*

Proof. Let L denote the set of leaves of T and let $T' = T - L$. Let (A, B) be a bipartition of T'. Let $\mathcal{P} = (c_t)_{t\in\mathbb{N}}$ be the R_2-process of $A \cup L$ on T. We prove that $\mathcal{P}$ converges to 1 by induction over the order of T. If T has order at most 2, then the result is trivial. Hence, we may assume that T has at least 3 vertices. If $v \in A \cup B$ is a leaf of T', then all but at most one of the neighbours of v belong to L and hence $c_2(v) = 1$. If $v \in A \cup B$ is not a leaf of T', then v has at least two neighbours in $A \cup B$ and hence $c_2(v) = 1 - c_1(v)$. Therefore, if L' denotes the set of leaves of T', then $(c_2^{-1}(1) \setminus (L \cup L'), c_2^{-1}(0) \setminus (L \cup L'))$ is a bipartition of $T' - L'$. By induction, the R_2-process $\left(c_{t+1}\ |_{V(T')}\right)_{t\in\mathbb{N}}$ on T' converges to 1. Hence also $\mathcal{P}$ converges to 1. □

Note that for a path P_n of order $n \geq 2$, Proposition 1 implies $r_2(P_n) \leq \left\lfloor \frac{n+2}{2} \right\rfloor$. In fact, the results of Section 2.2 easily imply $r_2(P_n) = \left\lfloor \frac{n+2}{2} \right\rfloor$, which means that Proposition 1 is best possible.

Increasing the number of vertices with initial label 1 does not always help the convergence. A simple example for this is a path $P : v_1v_2v_3v_4$ of order 4 with $f(v_1) = f(v_4) = 2$ and $f(v_2) = f(v_3) = 1$. While the R_f-process of $\{v_1, v_4\}$ on P converges, the R_f-process of $\{v_1, v_2, v_4\}$ on P does not.

2.1. The complexity of $\mathrm{r}_f(G)$

In this subsection, we consider the following decision problem.

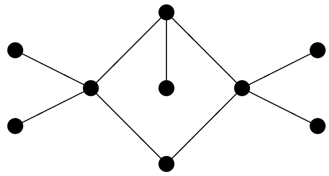

Fig. 1. Clause gadget $G(C_j)$.

R_k-Conversion Set
Instance: A graph G and an integer $c \geq 0$.
Question: $\mathrm{r}_k(G) \leq c$?

Note that a set of vertices is an R_1-conversion set of some graph if and only if it contains all vertices of the graph, that is, R_1-Conversion Set is trivial.

The proof of Theorem 2 illustrates that synchronicity is crucial for R_f-processes in the sense that 'things have to happen at the right moment'.

Theorem 2. R_2-Conversion Set *is NP-hard.*

Proof. We describe a reduction from Satisfiability. Let $\mathcal{C}$ be an instance of Satisfiability that uses the boolean variables $x_1, x_2, \ldots, x_n$ and consists of the clauses $C_1, C_2, \ldots, C_m$ such that for every $1 \leq i \leq n$, at most 3 clauses of $\mathcal{C}$ contain either x_i or $\bar{x}_i$, and for every $1 \leq j \leq m$, the clause C_j contains at most 3 literals. Note that the restriction of Satisfiability to such instances is still NP-complete (cf. [LO1] in [14]). Clearly, we may additionally assume that for every $1 \leq i \leq n$, some clause of $\mathcal{C}$ contains x_i and some clause of $\mathcal{C}$ contains $\bar{x}_i$. Consequently, for every $1 \leq i \leq n$, each of the two literals x_i and $\bar{x}_i$ is contained in exactly 1 or 2 of the clauses of $\mathcal{C}$.

We construct G and c as in R_2-Conversion Set such that the encoding length of (G, c) is polynomial in n and m, and $\mathcal{C}$ is satisfiable if and only if $\mathrm{r}_2(G) \leq c$. For every $1 \leq j \leq m$, we add to G a clause gadget $G(C_j)$ as in Fig. 1. For every $1 \leq i \leq n$, we add to G a variable gadget $G(x_i)$ as in Fig. 2. Furthermore, for every occurrence of some variable x_i (negated variable $\bar{x}_i$) in some clause C_j, we add an edge between a vertex in $\{x_{i,1}, x_{i,2}\}$ ($\{\bar{x}_{i,1}, \bar{x}_{i,2}\}$) and the clause vertex C_j such that the one or two edges corresponding to occurrences of x_i ($\bar{x}_i$) form a matching. Finally, we set $c = 38n + 5m$ (see Fig. 3).

First we assume that $\mathcal{C}$ is satisfiable and consider a satisfying truth assignment $\mathcal{A}$. Let $c_1 : V(G) \to \{0, 1\}$ be such that $c_1(v) = 1$ if and only if either v is a vertex of degree 1, or $v \in \{x_{i,1}, x_{i,2}\}$ for some $1 \leq i \leq n$ such that x_i is true in $\mathcal{A}$, or $v \in \{\bar{x}_{i,1}, \bar{x}_{i,2}\}$ for some $1 \leq i \leq n$ such that x_i is false in $\mathcal{A}$. Clearly, $|c_1^{-1}(1)| = c$. If x_i is true in $\mathcal{A}$ and x_i occurs in C_j, then

v	$x_{i,1}$	$u_{i,1}$	$u_{i,2}$	$u_{i,3}$	$\bar{x}_{i,1}$	$d_{i,1}$	$e_{i,1}$	$f_{i,1}$	C_j
$c_1(v)$	1	0	0	0	0	0	0	0	0
$c_2(v)$	0	1	1	0	1	1	1	0	1
$c_3(v)$	1	1	1	1	1	1	1	1	0 or 1
$c_4(v)$	1	1	1	1	1	1	1	1	1.

Since $\mathcal{A}$ is a satisfying truth assignment, we obtain, by symmetry, that $c_4^{-1}(1) = V(G)$, that is, $\mathrm{r}_2(G) \leq c$.

For the converse, we assume that $\mathcal{P} = (c_t)_{t \in \mathbb{N}}$ is an R_2-process on G that converges to 1 and satisfies $c_1(V(G)) \leq c$. Let $S_1 = c_1^{-1}(1)$. Clearly, S_1 contains all $36n + 5m$ vertices of degree 1 in G, that is, there are at most $2n$ further vertices in S_1. Note that, if $w_1w_2w_3w_4w_1$ is a cycle of length 4 in G, then the convergence of $\mathcal{P}$ implies that there is no $t \in \mathbb{N}$ with $c_t(w_1) = c_t(w_3) = 1 - c_t(w_2) = 1 - c_t(w_4)$. Therefore, for every $1 \leq i \leq n$, S_1 contains

- either a vertex in the cycle of length 4 that contains $d_{i,1}$ and $e_{i,1}$,
- or one of the two vertices $x_{i,1}$ and $\bar{x}_{i,1}$.

By symmetry and the order of S_1, this implies that for every $1 \leq i \leq n$, S_1 contains exactly two of the four vertices in $\{x_{i,1}, x_{i,2}, \bar{x}_{i,1}, \bar{x}_{i,2}\}$. In view of the cycle of length 4 that contains $x_{i,1}$ and $x_{i,2}$, we obtain, by symmetry, that for every $1 \leq i \leq n$, S_1 contains either $x_{i,1}$ and $x_{i,2}$, or $\bar{x}_{i,1}$ and $\bar{x}_{i,2}$. Hence $|S_1| = c$ and S_1 does not contain any vertex of degree more than 1 that belongs to a clause gadget. Furthermore, S_1 defines a truth assignment $\mathcal{A}$ in which the variable x_i for some $1 \leq i \leq n$ is set true if and only if $c_1(x_{i,1}) = 1$. In view of the cycle of length 4 in $G(C_j)$ for every $1 \leq j \leq m$, we obtain that $\mathcal{A}$ is a satisfying truth assignment for $\mathcal{C}$, which completes the proof. □

A question left open by Theorem 2 is whether R_k-Conversion Set actually lies in NP. One natural approach to prove this would require that every R_f-process that converges to 1, reaches the all-1-labelling after a polynomial number of steps. For some majority processes, this actually follows from general results as in [15,33,34] (cf. Corollary 2.3 in [26]). We will now argue why these results seem not to imply that R_k-Conversion Set is in NP for arbitrary k. In fact the arguments used in [15,33,34] rely on convexity properties and these are satisfied essentially only for majority processes. We substantiate this observation by formulating R_f-processes within the general framework proposed by Poljak and Turzík. In [33,34] Poljak and

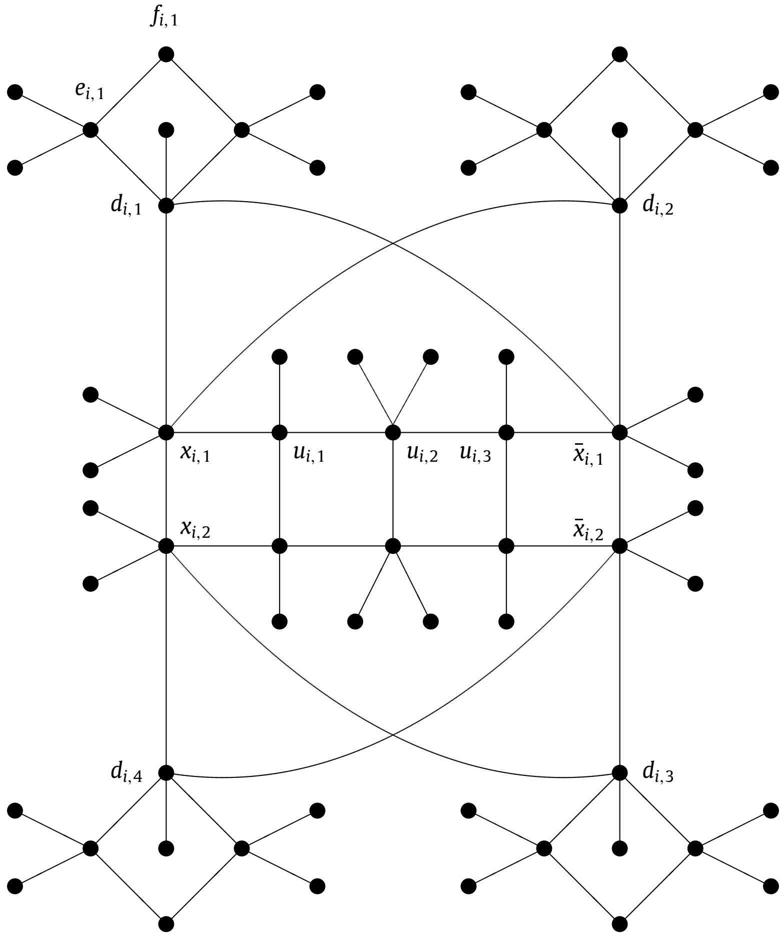

Fig. 2. Variable gadget $G(x_i)$.

Turzík consider a process of the form $(x_t)_{t\in\mathbb{N}}$ such that $x_1 \in \mathbb{Z}^m$ and for every $t \in \mathbb{N}$, $x_{t+1} = g(A \cdot x_t)$ where $A \in \mathbb{Z}^{m\times m}$ and $g : \mathbb{Z}^m \to \mathbb{Z}^m$. Under certain assumptions, they prove that $(x_t)_{t\in\mathbb{N}}$ has a period of length 1 or 2, and give an upper bound on the maximum number of steps needed to reach the periodic behaviour. It is possible to formulate R_f-processes within their framework as follows (cf. Theorem 2 in [34]).

Let $\mathcal{P} = (c_t)_{t\in\mathbb{N}}$ be an R_f-process on some graph G with vertex set $V(G) = \{v_1, v_2, \ldots, v_n\}$. For every $t \in \mathbb{N}$, let $x_t = (c_t(v_1), c_t(v_1), c_t(v_2), c_t(v_2), \ldots, c_t(v_n), c_t(v_n))^T$. Let $A = (a_{i,j})_{1\le i,j\le 2n} \in \{0, 1\}^{2n\times 2n}$ be such that

$$a_{i,j} = \begin{cases} 1, & \text{if } i = j \text{ and } i \text{ is odd, or } i \text{ and } j \text{ are even and } v_{\frac{i}{2}} v_{\frac{j}{2}} \in E(G), \\ 0 & \text{otherwise,} \end{cases}$$

that is, A is symmetric and arises by suitably combining the identity matrix with the adjacency matrix of G. By construction,

$$A \cdot x_t = (c_t(v_1), c_t(N_G(v_1)), c_t(v_2), c_t(N_G(v_2)), \ldots, c_t(v_n), c_t(N_G(v_n)))^T .$$

If for every $1 \le i \le n$, the function $g_i : \{0, 1\} \times \mathbb{Z} \to \left\{\binom{0}{0}, \binom{1}{1}\right\}$ is such that

$$g_i((x, y)) = \begin{cases} \binom{1}{1}, & \text{if either } x = 1 \text{ and } y \ge d_G(v_i) - f(v_i) + 1, \text{ or } x = 0 \text{ and } y \ge f(v_i), \\ \binom{0}{0}, & \text{if either } x = 1 \text{ and } y \le d_G(v_i) - f(v_i), \text{ or } x = 0 \text{ and } y \le f(v_i) - 1 \end{cases}$$

and $g : (\{0, 1\} \times \mathbb{Z})^n \to \mathbb{Z}^{2n}$ equals $g = g_1 \times g_2 \times \cdots \times g_n$, then $x_{t+1} = g(A \cdot x_t)$ for every $t \in \mathbb{N}$. Now, in order to apply Theorem 2 in [34], the functions g_i have to be *strongly cyclically monotonous*. It is easy to verify that this is the case if and only if $f(v) = \left\lfloor \frac{d_G(v)}{2} \right\rfloor + 1$ for every $v \in V(G)$, that is, if the R_f-process is a local majority process as considered by Mustafa and Pekeč [26].

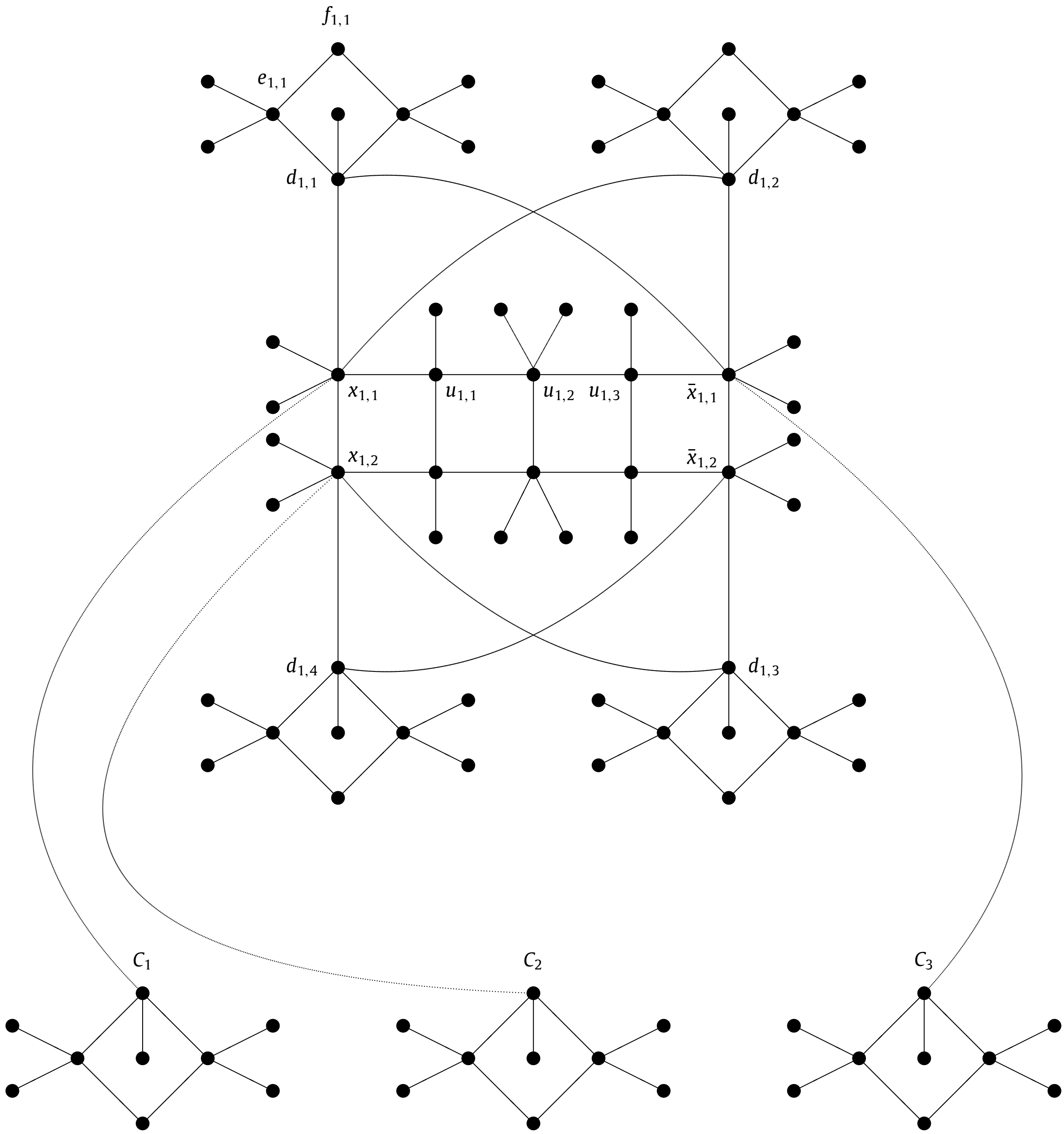

Fig. 3. An illustration of the variable gadget $G(x_1)$ and its connections to the clause gadgets $G(C_1)$, $G(C_2)$, and $G(C_3)$ assuming that x_1 is a literal in C_1 and C_2 and $\bar{x}_1$ is a literal in C_3.

2.2. Computing $\mathrm{r}_f(G)$ for certain paths and cycles

Computing $\mathrm{r}_f(G)$ is not easy even for graphs with a simple structure such as paths and cycles. Clearly, since all vertices have degree at most 2 in these graphs, vertices v with $f(v) \notin \{1, 2\}$ change their label either always or never and we can focus on vertices v with $f(v) \in \{1, 2\}$. A possible interpretation for the behaviour of vertices u with $f(u) = 1$ and v with $f(v) = 2$ within a cycle is that both apply a local majority rule but use different rules for tie breaking. While u changes its label already if half of its neighbours have a different label, v only does so in case of strict majority. Therefore, the results we present in this section can be considered as a case study of the behaviour of majority processes with non-uniform tie breaking.

The next lemma collects a number of useful properties of R_f-processes that converge to 1.

Lemma 3. *If G is a graph, $f : V(G) \to \mathbb{Z}$ is a function, and $\mathcal{P} = (c_t)_{t\in\mathbb{N}}$ is an R_f-process on G that converges to 1, then the following properties hold.*

(i) $f(v) \geq 1$ *for every vertex* $v \in V(G)$.
(ii) *If* $v \in V(G)$ *satisfies* $f(v) > d_G(v)$, *then* $c_t(v) = 1$ *for every* $t \in \mathbb{N}$.
(iii) *If* $uv \in E(G)$ *satisfies* $f(u) = f(v) = 1$, *then* $c_t(u) = c_t(v)$ *for every* $t \in \mathbb{N}$.
(iv) *If* $v_1v_2v_3$ *is a path in G that satisfies* $f(v_1) = f(v_2) = f(v_3) = 1$ *and* $d_G(v_2) = 2$, *then* $c_t(v) = 1$ *for every* $t \in \mathbb{N}$ *and every* $v \in N_G(v_1) \cup N_G(v_3)$.
(v) *If* $uv \in E(G)$ *satisfies* $f(u) = f(v) = 2$ *and* $d_G(u) = d_G(v) = 2$, *then* $(c_t(u), c_t(v)) \neq (0, 0)$ *for every* $t \in \mathbb{N}$. *Furthermore, if* $c_1(u) = c_1(v) = 1$, *then* $c_t(u) = c_t(v) = 1$ *for every* $t \in \mathbb{N}$.
(vi) *If* $v_1v_2v_3$ *is a path in G that satisfies* $f(v_1) = f(v_2) = 1$, $f(v_3) = 2$, *and* $d_G(v_2) = d_G(v_3) = 2$, *then* $(c_t(v_2), c_t(v_3)) \neq (0, 0)$ *for every* $t \in \mathbb{N}$. *Furthermore, if* $c_1(v_2) = c_1(v_3) = 1$, *then* $c_t(v_2) = c_t(v_3) = 1$ *for every* $t \in \mathbb{N}$.
(vii) *If* $u_ru_{r-1}\ldots u_2u_1v_1v_2\ldots v_{s-1}v_s$ *is a path in G that satisfies* $r, s \geq 2$, $f(u_1) = f(v_1) = 1$, $d_G(u_1) = d_G(v_1) = 2$, $f(u_i) = d_G(u_i) = 2$ *for every* $2 \leq i \leq r-2$, $f(v_i) = d_G(v_i) = 2$ *for every* $2 \leq i \leq s-2$, $c_t(u_r) = c_t(u_{r-1}) = c_t(v_s) = c_t(v_{s-1}) = 1$ *for every* $t \in \mathbb{N}$, $(c_1(u_i), c_1(u_{i+1})) \neq (1, 1)$ *for every* $1 \leq i \leq r-2$, *and* $(c_1(v_i), c_1(v_{i+1})) \neq (1, 1)$ *for every* $1 \leq i \leq s-2$, *then* $r = s$.
(viii) *If* $v_0v_1\ldots v_rv_{r+1}$ *is a path in G that satisfies* $r \geq 3$, $f(v_0) = f(v_1) = f(v_r) = f(v_{r+1}) = 1$, $f(v_i) = 2$ *for every* $2 \leq i \leq r-1$, $d_G(v_i) = 2$ *for every* $1 \leq i \leq r$, *then there is some i with* $1 \leq i \leq r-1$ *such that* $c_t(v_i) = c_t(v_{i+1}) = 1$ *for every* $t \in \mathbb{N}$.
Furthermore, if

$$j_{\min} = \min\{i \mid 1 \leq i \leq r-1 \text{ and } c_1(v_i) = c_1(v_{i+1}) = 1\} \text{ and}$$
$$j_{\max} = \max\{i \mid 1 \leq i \leq r-1 \text{ and } c_1(v_i) = c_1(v_{i+1}) = 1\},$$

then

$$c_1(\{v_i \mid 1 \leq i \leq j_{\min}-1\}) = \left\lfloor \frac{j_{\min}-1}{2} \right\rfloor,$$
$$c_1(\{v_i \mid j_{\max}+2 \leq i \leq r\}) = \left\lfloor \frac{r-j_{\max}-1}{2} \right\rfloor, \text{ and}$$
$$c_1(\{v_i \mid j_{\min}+2 \leq i \leq j_{\max}-1\}) \geq \left\lfloor \frac{j_{\max}-j_{\min}-2}{2} \right\rfloor.$$

Proof. (i) If there is some vertex $v \in V(G)$ with $f(v) \leq 0$, then $c_t(v) \neq c_{t+1}(v)$ for every $t \in \mathbb{N}$ which contradicts the convergence of $\mathcal{P}$.
(ii) If $v \in V(G)$ is a vertex with $f(v) > d_G(u)$, then $c_t(v) = c_{t+1}(v)$ for every $t \in \mathbb{N}$. Hence the convergence of $\mathcal{P}$ implies $c_t(v) = 1$ for every $t \in \mathbb{N}$.
(iii) If $uv \in E(G)$ is an edge with $f(u) = f(v) = 1$ and $c_t(u) \neq c_t(v)$ for some $t \in \mathbb{N}$, then $c_{t+1}(u) = 1 - c_t(u) \neq 1 - c_t(v) = c_{t+1}(v)$ and thus $c_s(u) \neq c_s(v)$ for every $s \in \mathbb{N}$ with $s \geq t$ by an inductive argument. Hence the convergence of $\mathcal{P}$ implies $c_t(u) = c_t(v)$ for every $t \in \mathbb{N}$.
(iv) Let the path $v_1v_2v_3$ be as in (iv). By (iii), we obtain $c_t(v_1) = c_t(v_2) = c_t(v_3)$ for every $t \in \mathbb{N}$. If $c_t(v_2) = 0$ for some $t \in \mathbb{N}$, then there is some $s \in \mathbb{N}$ with $s \geq t$ such that $c_s(v_2) = 0$ and $1 \in \{c_s(v_1), c_s(v_3)\}$. Hence $c_s(v_1) \neq c_s(v_2)$ or $c_s(v_2) \neq c_s(v_3)$, which is a contradiction. Hence $c_t(v_1) = c_t(v_2) = c_t(v_3) = 1$ for every $t \in \mathbb{N}$. This also implies $c_t(v) = 1$ for every $t \in \mathbb{N}$ and every $v \in N_G(v_1) \cup N_G(v_3)$.
(v) Let the edge uv be as in (v). If $c_t(u) = c_t(v) = 0$ for some $t \in \mathbb{N}$, then $c_s(u) = c_s(v) = 0$ for every $s \in \mathbb{N}$ with $s \geq t$, which contradicts the convergence of $\mathcal{P}$. Furthermore, if $c_1(u) = c_1(v) = 1$, then clearly $c_t(u) = c_t(v) = 1$ for every $t \in \mathbb{N}$.
(vi) Let the path $v_1v_2v_3$ be as in (vi). If $c_t(v_2) = c_t(v_3) = 0$ for some $t \in \mathbb{N}$, then (iii) implies $c_t(v_1) = 0$ and hence $c_{t+1}(v_2) = c_{t+1}(v_3) = 0$. By an inductive argument, this implies $c_s(v_2) = c_s(v_3) = 0$ for every $s \in \mathbb{N}$ with $s \geq t$, which contradicts the convergence of $\mathcal{P}$. Furthermore, if $c_1(v_2) = c_1(v_3) = 1$, then (iii) implies $c_1(v_1) = 1$ and hence $c_2(v_2) = c_2(v_3) = 1$. By an inductive argument, this implies $c_t(v_2) = c_t(v_3) = 1$ for every $t \in \mathbb{N}$.
(vii) Let the path $u_ru_{r-1}\ldots u_2u_1v_1v_2\ldots v_{s-1}v_s$ be as in (vii). By assumption, we obtain that $C_u = (c_1(u_{r-1}), c_1(u_{r-2}), \ldots, c_1(u_1))$ does not contain two consecutive 1-entries. By (v) and (vi), C_u does not contain two consecutive 0-entries. Therefore, C_u starts with a 1-entry and then the entries 0 and 1 alternate, that is, $C_u = (1, 0, 1, 0, \ldots)$. By symmetry, $C_v = (c_1(v_{s-1}), c_1(v_{s-2}), \ldots, c_1(v_1))$ starts with a 1-entry and then the entries 0 and 1 alternate. By (iii), this implies that r and s have the same parity modulo 2. For contradiction, we assume that $r - s > 0$ and that the process $\mathcal{P}$ and the path $u_r\ldots u_1v_1\ldots v_s$ are chosen such that $r + s \geq 2$ is minimum possible. If $s = 2$, then $(c_1(u_2), c_1(u_1), c_1(v_1), c_1(v_2)) = (0, 1, 1, 1)$ and hence $(c_2(u_1), c_2(v_1)) = (0, 1)$, which contradicts (iii). Hence $s \geq 3$ and therefore $r \geq 5$. In view of the properties of C_u and C_v noted above, we obtain $c_2(w) = 1 - c_1(w)$ for every $w \in \{u_{r-2}, \ldots, u_1, v_1, \ldots, v_{s-2}\}$. Now the convergent process $(c_{t+1})_{t\in\mathbb{N}}$ and the path $u_{r-1}u_{r-2}\ldots u_2u_1v_1v_2\ldots v_{s-2}v_{s-1}$ yield a counterexample with $(r-1)+(s-1) < r + s$ which is a contradiction.

(viii) Let the path $v_0v_1\dots v_rv_{r+1}$ be as in (viii). Note that, by (iii), $c_t(v_0)=c_t(v_1)$ and $c_t(v_r)=c_t(v_{r+1})$ for every $t\in\mathbb{N}$. By (v) and (vi), $C=(c_1(v_1),c_1(v_2),\dots,c_1(v_r))$ does not contain two consecutive 0-entries. For contradiction, we assume that C does not contain two consecutive 1-entries. This implies that in C the entries 0 and 1 alternate. We obtain $c_2(v_i)=1-c_1(v_i)$ and $c_3(v_i)=1-c_2(v_i)=c_1(v_i)$ for every $0\le i\le r+1$, which contradicts the convergence of $\mathcal{P}$. Therefore there is some i with $1\le i\le r-1$ such that $c_1(v_i)=c_1(v_{i+1})=1$. By (v) and (vi), this implies $c_t(v_i)=c_t(v_{i+1})=1$ for every $t\in\mathbb{N}$. Note that $j_{\min}$ and $j_{\max}$ are well-defined. By the definition of $j_{\min}$, $C_1=(c_1(v_{j_{\min}-1}),c_1(v_{j_{\min}-2}),\dots,c_1(v_1))$ starts with a 0-entry and then the entries 0 and 1 alternate. Hence the number of 1-entries in C_1 is exactly $\left\lfloor\frac{j_{\min}-1}{2}\right\rfloor$. Similarly, the number of 1-entries in $(c_1(v_{j_{\max}+2}),c_1(v_{j_{\max}+3}),\dots,c_1(v_r))$ is exactly $\left\lfloor\frac{r-j_{\max}-1}{2}\right\rfloor$. Finally, since $C_2=(c_1(v_{j_{\min}+2}),\dots,c_1(v_{j_{\max}-1}))$ does not contain two consecutive 0-entries, the number of 1-entries in C_2 is at least $\left\lfloor\frac{j_{\max}-j_{\min}-2}{2}\right\rfloor$. This completes the proof. □

While Lemma 3 essentially serves to derive a lower bound on $\mathrm{r}_f(G)$, the following two lemmas serve to show that this lower bound can be attained.

Lemma 4. *If $P: u_ru_{r-1}\dots u_2u_1v_1v_2\dots v_{r-1}v_r$ is a path, $f: V(P)\to\mathbb{Z}$ is a function, and $c_1: V(P)\to\{0,1\}$ is a labelling such that $r\ge 2$, $f(u_1)=f(v_1)=1$, $f(u_i)=f(v_i)=2$ for every $2\le i\le r$, $c_1(u_r)=c_1(u_{r-1})=c_1(v_r)=c_1(v_{r-1})=1$, and $C_u=(c_1(u_1),c_1(u_2),\dots,c_1(u_{r-1}))$ as well as $C_v=(c_1(v_1),c_1(v_2),\dots,c_1(v_{r-1}))$ contain no two consecutive entries that are equal, then the R_f-process $\mathcal{P}=(c_t)_{t\in\mathbb{N}}$ converges to 1.*

Proof. We prove the statement by induction on r. For $r=2$, the result is trivial. Now let $r\ge 3$. Note that $c_1(u_r)=c_1(u_{r-1})=c_1(v_r)=c_1(v_{r-1})=1$ and $f(u_r)=f(u_{r-1})=f(v_r)=f(v_{r-1})=2$ implies $c_t(u_r)=c_t(u_{r-1})=c_t(v_r)=c_t(v_{r-1})=1$ for every $t\in\mathbb{N}$. The structure of C_u and C_v implies that $c_1(u_i)=c_1(v_i)$ for $1\le i\le r$ and

$$c_2(u_i)=c_2(v_i)=\begin{cases}1, & i\in\{r-2,r-1,r\}\\ 1-c_1(u_i), & 1\le i\le r-3.\end{cases}$$

Now the result follows immediately by induction applied to the path $u_{r-1}\dots u_1v_1\dots v_{r-1}$. □

Lemma 5. *If $P: v_1v_2\dots v_n$ is a path, $f: V(P)\to\mathbb{Z}$ is a function, and $c_1: V(P)\to\{0,1\}$ is a labelling such that $f(v)=2$ for every $v\in V(P)$, $c_1(v_1)=c_1(v_n)=1$, and $C=(c_1(v_1),c_1(v_2),\dots,c_1(v_n))$ does not contain two consecutive 0-entries, then the R_f-process $\mathcal{P}=(c_t)_{t\in\mathbb{N}}$ converges to 1. Furthermore, $\mathrm{r}_f(P)=\left\lceil\frac{n+1}{2}\right\rceil$.*

Proof. We prove the first statement by induction on n. For $n\le 3$, the result is trivial. Hence, we assume $n\ge 4$. If C contains two consecutive 1-entries, say $c_1(v_i)=c_1(v_{i+1})=1$, then $c_t(v_i)=c_t(v_{i+1})=1$ for every $t\in\mathbb{N}$ and the result follows by induction applied to the two paths $v_1\dots v_i$ and $v_{i+1}\dots v_n$. Hence we may assume that C starts and ends with a 1-entry and that the entries 0 and 1 alternate, that is, $C=(1,0,1,0,1,\dots,0,1,0,1)$. We obtain $(c_2(v_1),c_2(v_3),\dots,c_2(v_n))=(1,1,0,1,0,\dots,1,0,1,1)$ and the result follows by induction applied to the path $v_2\dots v_{n-1}$. This completes the proof of the first statement, which already implies $\mathrm{r}_f(P)\le\left\lceil\frac{n+1}{2}\right\rceil$. The remaining estimate $\mathrm{r}_f(P)\ge\left\lceil\frac{n+1}{2}\right\rceil$ follows easily from (ii) and (v) of Lemma 3. □

Lemmas 3 to 5 suffice to describe an efficient algorithm which calculates $r_f(G)$ for a path or cycle G and functions $f: V(G)\to\mathbb{Z}$ such that $f(v)\ge 1$ for every $v\in V(G)$ and, for every $v\in V(G)$ with $f(v)=1$, there is some neighbour $u\in N_G(v)$ with $f(u)=1$. In what follows we refer to such a pair (G,f) as a *restricted path instance* or a *restricted cycle instance*, respectively. We describe this algorithm in detail for restricted path instances and then explain the few and simple modifications for restricted cycle instances. The key property used by this algorithm is that consecutive 1-values of f essentially allow us to split the instance between the two 1-values into smaller and simpler parts. The arising smaller parts have to converge in some sense on their own sake and there are simple synchronicity conditions on their behaviour that are necessary for overall convergence.

Therefore, let (P,f) be a restricted path instance. Let $P: v_1v_2\dots v_n$. First we describe four simple reduction operations.

(O_1) If there is some vertex $v_i\in V(P)$ with $f(v_i)\ge 3$ and either $i=1$ or $i=n$, then let

$$f'(v)=\begin{cases}2, & v=v_i,\\ f(v), & v\in V(P)\setminus\{v_i\}.\end{cases}$$

Since $\mathcal{Q}$ is an R_f-process on P if and only if $\mathcal{Q}$ is an $\mathrm{R}_{f'}$-process on P, we obtain $\mathrm{r}_f(P)=\mathrm{r}_{f'}(P)$. In this case, the algorithm recurses over the restricted path instance (P,f').

(O_2) If there is some vertex $v_i\in V(P)$ with $f(v_i)\ge 3$ and $2\le i\le n-1$, then let

$$\begin{aligned}&P_1: v_1\dots v_i \quad\text{and}\quad f_1=f\mid_{V(P_1)}\\ &P_2: v_i\dots v_n \quad\text{and}\quad f_2=f\mid_{V(P_2)}.\end{aligned}$$

By (ii) of Lemma 3, $\mathrm{r}_f(P)=\mathrm{r}_{f_1}(P_1)+\mathrm{r}_{f_2}(P_2)-1$. In this case, the algorithm recurses over the two restricted path instances (P_1,f_1) and (P_2,f_2).

(O_3) If there are two indices r and s with $1 \le r < s \le n$ such that $s - r + 1 \ge 3$ and $f(v_i) = 1$ for every $r \le i \le s$, then we may assume that either $r = 1$ or $r > 1$ and $f(v_{r-1}) \ge 2$, and either $s = n$ or $s < n$ and $f(v_{s+1}) \ge 2$. Let

$$P_1 : v_1 \dots v_{r-1} \quad \text{and} \quad f_1 = f\mid_{V(P_1)},$$
$$P_2 : v_{s+1} \dots v_n \quad \text{and} \quad f_2 = f\mid_{V(P_2)}.$$

By (ii) and (iv) of Lemma 3, $\mathrm{r}_f(P) = \mathrm{r}_{f_1}(P_1) + \mathrm{r}_{f_2}(P_2) + (s - r + 1)$. In this case, the algorithm recurses over the two restricted path instances (P_1, f_1) and (P_2, f_2).

(O_4) If $f(v_1) = f(v_2) = 1$, and $f(v_3) \ge 2$, then let $P' : v_3 \dots v_n$ and $f' = f\mid_{V(P')}$. (ii) and (iii) of Lemma 3 easily imply $\mathrm{r}_f(P) = \mathrm{r}_{f'}(P') + 2$. In this case, the algorithm recurses over the restricted path instance (P', f'). A similar operation applies in the case that $f(v_n) = f(v_{n-1}) = 1$, and $f(v_{n-2}) \ge 2$.

Applying the operations (O_1) to (O_4) as often as possible results in polynomially many instances of polynomial encoding length with additional properties, which can all be solved independently. Therefore, we may now assume that $f(v) \in \{1, 2\}$ for all $v \in V(P)$, $f(v_1) = f(v_n) = 2$, and $(f(v_1), f(v_2), \dots, f(v_n))$ does not contain three consecutive 1-entries. Furthermore, in view of Lemma 5, we may assume there is at least one vertex $v \in V(P)$ with $f(v) = 1$. We call a restricted path instance that satisfies these additional conditions *reduced*.

For every two consecutive vertices v_i and v_{i+1} with $f(v_i) = f(v_{i+1}) = 1$, we split P between v_i and v_{i+1}. This results in a decomposition

$$P : P_1 P_2 \dots P_r \tag{1}$$

of P such that

$$P_i : v_{i,1} \dots v_{i,n_i} \tag{2}$$

for $1 \le i \le r$ and some n_i,

$$\begin{aligned}
&(f(v_{1,1}), f(v_{1,2}), \dots, f(v_{1,n_1})) = (2, 2, \dots, 2, 1),\\
&(f(v_{i,1}), f(v_{i,2}), \dots, f(v_{i,n_i})) = (1, 2, \dots, 2, 1) \text{ for every } 2 \le i \le r-1, \text{ and}\\
&(f(v_{r,1}), f(v_{r,2}), \dots, f(v_{r,n_r})) = (1, 2, \dots, 2, 2).
\end{aligned} \tag{3}$$

Note that $n_1, n_r \ge 2$, $n_i \ge 3$ for every $2 \le i \le r-1$, and $n_1 + n_2 + \dots + n_r = n$.

For some $1 \le s \le r$, we call two integers $j_{s,\min}$ and $j_{s,\max}$ *admissible indices* if

$$\begin{array}{lll}
0 \le j_{s,\min} = j_{s,\max} \le n_1 - 1, & \text{if } s = 1\\
1 \le j_{s,\min} \le j_{s,\max} \le n_i - 1, & \text{if } 2 \le s \le r-1, \text{ and}\\
1 \le j_{s,\min} = j_{s,\max} \le n_r, & \text{if } s = r.
\end{array}$$

(Note that whenever we speak of admissible indices there is some reduced restricted path instance we implicitly refer to.)

Lemma 6. *Let (P, f) be a reduced restricted path instance. Let $P : P_1 P_2 \dots P_r$ be as in (1), (2), and (3).*

If $c_1 : V(P) \to \{0, 1\}$ is such that the R_f-process $\mathcal{P} = (c_t)_{t\in\mathbb{N}}$ converges to 1, then the indices $j_{i,\min}$ and $j_{i,\max}$ for every $1 \le i \le r$ with

$$j_{1,\min} = j_{1,\max} = \max\{0, \max\{j \mid 1 \le j \le n_1 - 1 \text{ and } c_1(v_{1,j}) = c_1(v_{1,j+1}) = 1\}\} \tag{4}$$
$$j_{i,\min} = \min\{j \mid 1 \le j \le n_i - 1 \text{ and } c_1(v_{i,j}) = c_1(v_{i,j+1}) = 1\} \text{ for every } 2 \le i \le r-1, \tag{5}$$
$$j_{i,\max} = \max\{j \mid 1 \le j \le n_i - 1 \text{ and } c_1(v_{i,j}) = c_1(v_{i,j+1}) = 1\} \text{ for every } 2 \le i \le r-1, \tag{6}$$
$$j_{r,\min} = j_{r,\max} = \min\{n_r, \min\{j \mid 1 \le j \le n_r - 1 \text{ and } c_1(v_{r,j}) = c_1(v_{r,j+1}) = 1\}\} \tag{7}$$

are well-defined, admissible, and satisfy

$$c_1(V(P_1)) \ge \begin{cases} 1 + \left\lfloor \frac{n_1 - 1}{2} \right\rfloor, & j_{1,\max} = 0,\\ 2 + \left\lfloor \frac{n_1 - 2}{2} \right\rfloor, & j_{1,\max} = 1,\\ 3 + \left\lfloor \frac{j_{1,\max} - 2}{2} \right\rfloor + \left\lfloor \frac{n_1 - j_{1,\max} - 1}{2} \right\rfloor, & j_{1,\max} \ge 2, \end{cases} \tag{8}$$

$$c_1(V(P_i)) \ge \begin{cases} 4 + \left\lfloor \frac{j_{i,\min} - 1}{2} \right\rfloor + \left\lfloor \frac{j_{i,\max} - j_{i,\min} - 2}{2} \right\rfloor + \left\lfloor \frac{n_i - j_{i,\max} - 1}{2} \right\rfloor, & j_{i,\max} - j_{i,\min} \ge 2,\\ 3 + \left\lfloor \frac{j_{i,\min} - 1}{2} \right\rfloor + \left\lfloor \frac{n_i - j_{i,\max} - 1}{2} \right\rfloor, & j_{i,\max} - j_{i,\min} = 1,\\ 2 + \left\lfloor \frac{j_{i,\min} - 1}{2} \right\rfloor + \left\lfloor \frac{n_i - j_{i,\max} - 1}{2} \right\rfloor, & j_{i,\max} - j_{i,\min} = 0 \end{cases}$$
$$\textit{for every } 2 \le i \le r-1, \tag{9}$$

$$c_1(V(P_r)) \geq \begin{cases} 1+\left\lfloor \frac{n_r-1}{2} \right\rfloor, & j_{r,\min}=n_r, \\ 2+\left\lfloor \frac{n_r-2}{2} \right\rfloor, & j_{r,\min}=n_r-1, \\ 3+\left\lfloor \frac{j_{r,\min}-1}{2} \right\rfloor + \left\lfloor \frac{n_r-j_{r,\min}-2}{2} \right\rfloor, & j_{r,\min}\leq n_r-2, \end{cases} \tag{10}$$

and

$$j_{i+1,\min} = n_i - j_{i,\max} \text{ for every } 1 \leq i \leq r-1. \tag{11}$$

Conversely, if there are admissible indices $j_{i,\min}$ and $j_{i,\max}$ for every $1 \leq i \leq r$ such that (11) is satisfied, then there is some $c_1 : V(G) \to \{0, 1\}$ that satisfies (4) to (7) and satisfies (8) to (10) with equality such that the R_f-process $\mathscr{P} = (c_t)_{t\in\mathbb{N}}$ converges to 1.

Proof. Let $c_1 : V(G) \to \{0, 1\}$ be such that the R_f-process $\mathscr{P} = (c_t)_{t\in\mathbb{N}}$ converges to 1. By (viii) of Lemma 3, for every $2 \leq i \leq r-1$, $(c_1(v_{i,1}), c_1(v_{i,2}), \ldots, c_1(v_{i,n_i}))$ contains two consecutive 1-entries. Therefore, for $2 \leq i \leq r-1$, the indices $j_{i,\min}$ and $j_{i,\max}$ as in (5) and (6) are well-defined. Furthermore, let $j_{1,\min} = j_{1,\max}$ and $j_{r,\min} = j_{r,\max}$ be as in (4) and (7). Note that these last four indices are always well-defined and that all indices are admissible. By (viii) Lemma 3, $c_1(V(P_i))$ satisfies (9). Furthermore, the definition of $j_{1,\max}$ and $j_{r,\min}$ and (v) of Lemma 3 imply that $c_1(V(P_1))$ satisfies (8) and that $c_1(V(P_r))$ satisfies (10). Finally, by (vii) of Lemma 3, the indices $j_{i,\max}$ and $j_{i+1,\min}$ satisfy (11).

Conversely, let $j_{i,\min}$ and $j_{i,\max}$ for every $1 \leq i \leq r$ be admissible indices such that (11) is satisfied. We construct some $c_1 : V(G) \to \{0, 1\}$ as follows. We set $c_1(v_{1,1}) = c_1(v_{r,n_r}) = 1$. For $2 \leq i \leq r-1$, we set $c_1(v_{i,j_{i,\min}}) = c_1(v_{i,j_{i,\min}+1}) = 1$ and $c_1(v_{i,j_{i,\max}}) = c_1(v_{i,j_{i,\max}+1}) = 1$. If $j_{1,\min} = j_{1,\max} \geq 1$, we proceed similarly for $i = 1$. If $j_{r,\min} = j_{r,\max} \leq n_r - 1$, we proceed similarly for $i = r$. Note that all values that we have set so far will never change during the process. This is obvious for $c_1(v_{1,1})$, $c_1(v_{r,n_r})$, and all consecutive 1-values $c_1(v_i) = c_1(v_{i+1}) = 1$ with $f(v_i) = f(v_{i+1}) = 2$. By (11), all remaining consecutive 1-values are part of a sequence of four consecutive 1-values $(c_1(v_i), c_1(v_{i+1}), c_1(v_{i+2}), c_1(v_{i+3})) = (1, 1, 1, 1)$ with $(f(v_i), f(v_{i+1}), f(v_{i+2}), f(v_{i+3})) = (2, 1, 1, 2)$. Hence also these values will never change during the process. We set the values $c_1(v)$ for vertices v on P that lie between $v_{i,j_{i,\max}}$ and $v_{i+1,j_{i+1,\min}}$ for some $1 \leq i \leq r-1$ exactly as in Lemma 4. Finally, we set all values $c_1(v)$ that are yet undefined using the minimum possible number of 1-values avoiding two consecutive 0-values. Clearly, this leads to some $c_1 : V(G) \to \{0, 1\}$ that satisfies (4) to (7) and satisfies (8) to (10) with equality. Now Lemmas 4 and 5 imply that the R_f-process $\mathscr{P} = (c_t)_{t\in\mathbb{N}}$ converges to 1, which completes the proof. □

Theorem 7. *There is a polynomial time algorithm that determines $\mathrm{r}_f(P)$ for restricted path instances (P, f).*

Proof. In view of the operations (O_1) to (O_4), it suffices to prove the result for reduced restricted path instances. Therefore, let (P, f) be such an instance. Let $P : v_1v_2 \ldots v_n$ and let $P : P_1P_2 \ldots P_r$ be as in (1) to (3). Our approach relies on dynamic programming.

Given admissible indices $j_{i,\min}$ and $j_{i,\max}$ for every $1 \leq i \leq s$ for some $1 \leq s \leq r$ that satisfy

$$j_{i+1,\min} = n_i - j_{i,\max} \text{ for every } 1 \leq i \leq s-1,$$

we call $((j_{1,\min}, j_{1,\max}), \ldots, (j_{s,\min}, j_{s,\max}))$ an *admissible index sequence*. Let the *cost* of an admissible index sequence $((j_{1,\min}, j_{1,\max}), \ldots, (j_{s,\min}, j_{s,\max}))$ equal $c_1(V(P_1)) + \cdots + c_1(V(P_s))$ where the values $c_1(V(P_1)), \ldots, c_1(V(P_s))$ satisfy the first s inequalities among (8) to (10) with equality. For some $1 \leq s \leq r$ and admissible indices $j_{s,\min}$ and $j_{s,\max}$ let $\mathrm{mincost}(s, j_{s,\min}, j_{s,\max})$ denote the minimum cost of some admissible index sequence $((j_{1,\min}, j_{1,\max}), \ldots, (j_{s,\min}, j_{s,\max}))$ or ∞ if no such admissible index sequence exists. Clearly, it is possible to determine the values of $\mathrm{mincost}(1, j_{1,\min}, j_{1,\max})$ for all admissible indices $j_{1,\min}$ and $j_{1,\max}$ using (8). Furthermore, given the values of $\mathrm{mincost}(s, j_{s,\min}, j_{s,\max})$ for some $1 \leq s < r$ and all admissible indices $j_{s,\min}$ and $j_{s,\max}$, it is possible to determine $\mathrm{mincost}(s+1, j_{s+1,\min}, j_{s+1,\max})$ for all admissible indices $j_{s+1,\min}$ and $j_{s+1,\max}$ using (11) and (8) to (10). By Lemma 6,

$$\mathrm{r}_f(P) = \min\left\{\mathrm{mincost}(r, j_{r,\min}, j_{r,\max}) \mid j_{r,\min} \text{ and } j_{r,\max} \text{ are admissible indices}\right\}.$$

The running time of the described dynamic programming procedure is clearly polynomial in the order of P, which completes the proof. □

Corollary 8. *There is a polynomial time algorithm that determines $\mathrm{r}_f(C)$ for restricted cycle instances (C, f).*

Proof. Let (C, f) be a restricted cycle instance. If there is a vertex $u \in V(C)$ with $f(u) \geq 3$ or there are three cyclically consecutive vertices u_1, u_2, and u_3 with $f(u_1) = f(u_2) = f(u_3) = 1$, then operations similar to (O_1) and (O_3) apply and the problem reduces to restricted path instances. If $f(u) = 2$ for every $u \in V(C)$, then it is an easy exercise to show $\mathrm{r}_f(C) = \left\lfloor \frac{|V(C)|}{2} \right\rfloor + 1$. Hence, we may assume that (C, f) satisfies analogous conditions as a reduced restricted path instance. This implies that we can decompose C in a similar way as such instances (cf. (1) to (3)) and determine $\mathrm{r}_f(C)$ using dynamic programming as in the proof of Theorem 7. Since condition (11) has to be satisfied cyclically, the dynamic programming has to maintain information concerning admissible index positions in the first and the last segment instead of just in the last segment. □

It is easy to see that the dynamic programming procedures described in Theorem 7 and Corollary 8 have quadratic running time. While it would be interesting to improve their running time, the first objectives for further research on algorithms in this context should be to eliminate the drawback of only working for restricted instances and to find – if possible – a simpler and more general approach, which would allow to efficiently solve the problem for larger classes of graphs.

Acknowledgements

The authors acknowledge partial support by the CAPES/DAAD Probral project ''Cycles, Convexity, and Searching in Graphs''.

References

[1] Z. Agur, Fixed points of majority rule cellular automata with application to plasticity and precision of the immune system, Complex Systems 5 (1991) 351–357.
[2] J.-P. Allouche, M. Courbage, G. Skordev, Notes on cellular automata, Cubo Mat. Educ. 3 (2001) 213–244.
[3] P. Balister, B. Bollobás, J.R. Johnson, M. Walters, Random majority percolation, Random Structures Algorithms 36 (2010) 315–340.
[4] C.L. Barrett, H.B. Hunt III, M.V. Marathe, S.S. Ravi, D.J. Rosenkrantz, R.E. Stearns, Complexity of reachability problems for finite discrete dynamical systems, J. Comput. System Sci. 72 (2006) 1317–1345.
[5] E. Berger, Dynamic monopolies of constant size, J. Combin. Theory Ser. B 83 (2001) 191–200.
[6] J.-C. Bermond, J. Bond, D. Peleg, S. Perennes, The power of small coalitions in graphs, Discrete Appl. Math. 127 (2003) 399–414.
[7] C.C. Centeno, M.C. Dourado, L. Draque Penso, D. Rautenbach, J.L. Szwarcfiter, Irreversible conversion of graphs, Theoret. Comput. Sci. 412 (2011) 3693–3700.
[8] N. Chen, On the approximability of influence in social networks, SIAM J. Discrete Math. 23 (2009) 1400–1415.
[9] P. Domingos, M. Richardson, Mining the network value of customers, in: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2001, pp. 57–66.
[10] P.A. Dreyer Jr., F.S. Roberts, Irreversible *k*-threshold processes: graph-theoretical threshold models of the spread of disease and of opinion, Discrete Appl. Math. 157 (2009) 1615–1627.
[11] P. Flocchini, F. Geurts, N. Santoro, Optimal irreversible dynamos in chordal rings, Discrete Appl. Math. 113 (2001) 23–42.
[12] P. Flocchini, R. Královič, A. Roncato, P. Ružička, N. Santoro, On time versus size for monotone dynamic monopolies in regular topologies, J. Discrete Algorithms 1 (2003) 129–150.
[13] J. French, A formal theory of social power, Psychology Rev. 63 (1956) 181–194.
[14] M.R. Garey, D.S. Johnson, Computers and Intractability, in: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, 1979.
[15] E. Goles, J. Olivos, Comportement périodique des fonctions à seuil binaires et applications, Discrete Appl. Math. 3 (1981) 93–105.
[16] Y. Hassin, D. Peleg, Distributed probabilistic polling and applications to proportionate agreement, Inf. Comput. 171 (2001) 248–268.
[17] S. Huang, Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery, J. Mol. Med. 77 (1999) 469–480.
[18] D. Kempe, J. Kleinberg, É Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, 137–146.
[19] R. Královič, On majority voting games in trees, Lect. Notes Comput. Sci. 2234 (2001) 282–291.
[20] S. Kutten, D. Peleg, Fault-local distributed mending, J. Algorithms 30 (1999) 144–165.
[21] N. Linial, D. Peleg, Y. Rabinovich, M.E. Saks, Sphere packing and local majorities in graphs, in: Second Israel Symposium on Theory of Computing Systems, ISTCS 1993, pp. 141–149.
[22] S. Maity, A. Nayak, B.K. Roy, Characterization of catastrophic faults in two-dimensional reconfigurable systolic arrays with unidirectional links, Inf. Process. Lett. 92 (2004) 189–197.
[23] A. Montanari, A. Saberi, Convergence to equilibrium in local interaction games, in: FOCS 2009, pp. 303–312.
[24] G. Moran, On the period-two-property of the majority operator in infinite graphs, Trans. Amer. Math. Soc. 347 (1995) 1649–1667.
[25] S. Morris, Contagion, Rev. Econom. Stud. 67 (2000) 57–78.
[26] N.H. Mustafa, A. Pekeč, Listen to your neighbors: how (not) to reach a consensus, SIAM J. Discrete. Math. 17 (2004) 634–660.
[27] T. Nakata, H. Imahayashi, M. Yamashita, A probabilistic local polling game on weighted directed graphs with an application to the distributed agreement problem, Networks 35 (2000) 266–273.
[28] A. Nayak, J. Ren, N. Santoro, An improved testing scheme for catastrophic fault patterns, Inf. Process. Lett. 73 (2000) 199–206.
[29] P. Orponen, Computational complexity of neural networks: A survey, Nordic J. Comput. 1 (1994) 94–110.
[30] D. Peleg, Local majorities, coalitions and monopolies in graphs: A review, Theor. Comput. Sci. 282 (2002) 231–257.
[31] D. Peleg, Size bounds for dynamic monopolies, Discrete Appl. Math. 86 (1998) 263–273.q.
[32] S. Poljak, M. Sura, On periodical behaviour in societies with symmetric influences, Combinatorica 3 (1983) 119–121.
[33] S. Poljak, D. Turzík, On an application of convexity to discrete systems, Discrete Appl. Math. 13 (1986) 27–32.
[34] S. Poljak, D. Turzík, On pre-periods of discrete influence systems, Discrete Appl. Math. 13 (1986) 33–39.
[35] J. Sima, P. Orponen, General-purpose computation with neural networks: a survey of complexity theoretic results, Neural Comput. 15 (2003) 2727–2778.